Jet Propulsion Laboratory
California Institute of Technology

Cloud-based Data Processing and Workflow Systems

Namrata Malarout

Scientific Applications Software Engineer

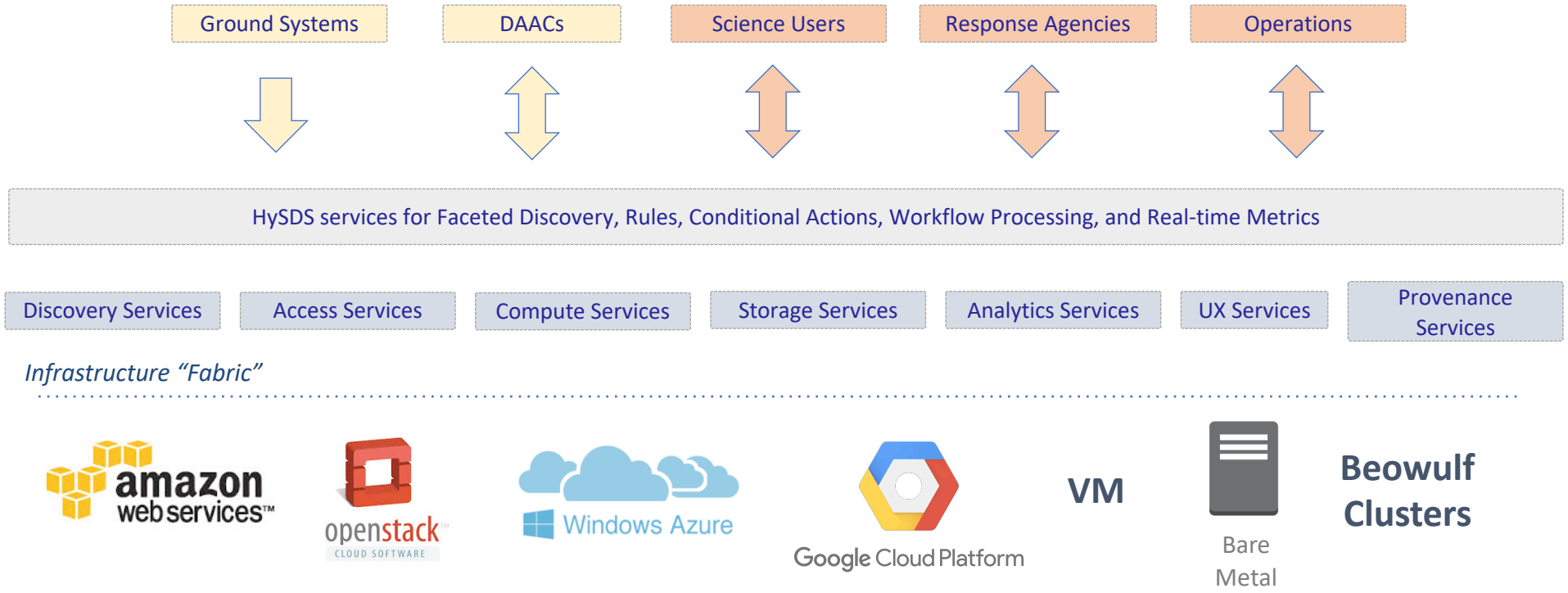
Jet Propulsion Laboratory

Brief Introduction to HySDS

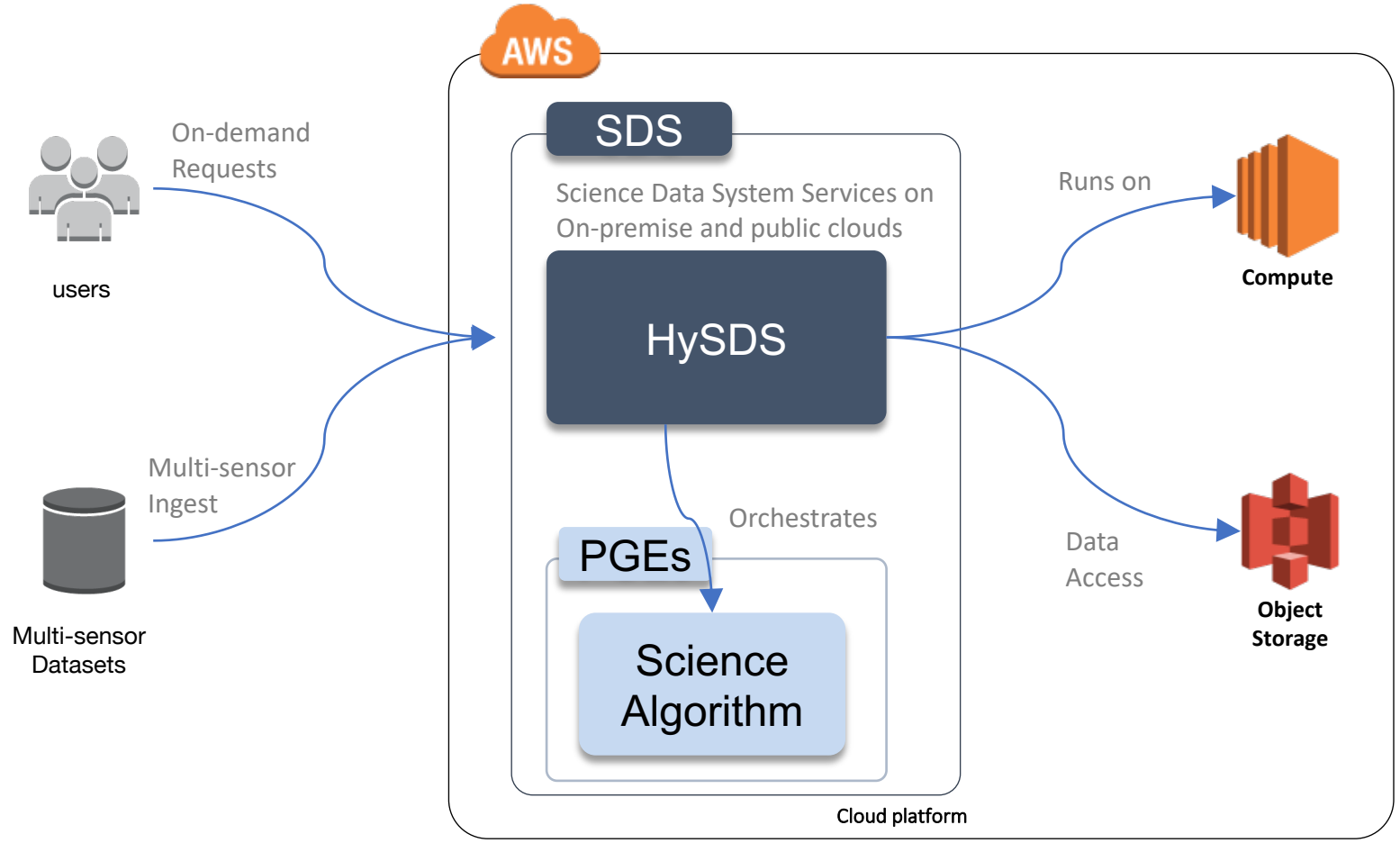
- Hybrid-cloud Science Data System
- SDS Services
  - Resource management, Data discovery, Workflow, UI interfaces, API interfaces
- Designed to run on public and on-premise clouds, as well as compute on legacy machines.
  - Mainly AWS and OpenStack
  - Supports cost-effective AWS *spot market*
- Resiliency and fault-tolerant compute
  - Useful at large scales and in spot market terminations
- Faceted rules for triggering and on-demand use
- Containerized components
  - Docker Containers as PGEs
  - Continuous Integration for Docker PGE deployment

Platform Portability

- Computing fabric over heterogeneous infrastructure
- Addresses vendor lock-in



Top-Level Software Components



Auto-Scaling Science Data System

- Auto Scaling group policies
  - scaling size
  - rest period
- Metric alarms
  - E.g. condition: work queue size > 20

Deployment Procedure

1. SA Team provisions AWS account and resources
   - Configure security
   - Create IAM accounts/roles
   - Configure VPC
   - Create buckets
   - Build AMIs

2. Deploy Engineer runs **terraform** to standup PCM instances

3. Deploy Engineer runs **sdscli** to configure HySDS cluster& apply adaption

HySDS Adaptions

- Advanced Rapid Imaging and Analysis (ARIA)
- MEaSUREs WVCC A-Train data fusion
- OCO-2 L2 Full Physics processing
- Getting Ready For NISAR (GRFN)
- SMAP in the Cloud (in progress)
- NISAR SDS (in progress)
- SWOT SDS (in progress)
- Second upcoming reprocessing campaign of OCO-2
- ARIA: Machine Learning of SAR

Concurrent Analysis Pipelines

- Concurrently keep up with:
  - Monitoring (real-time data)
  - Historic data analysis
  - Urgent response
  - On-demand analysis

Workflows: Explicit vs. Implicit

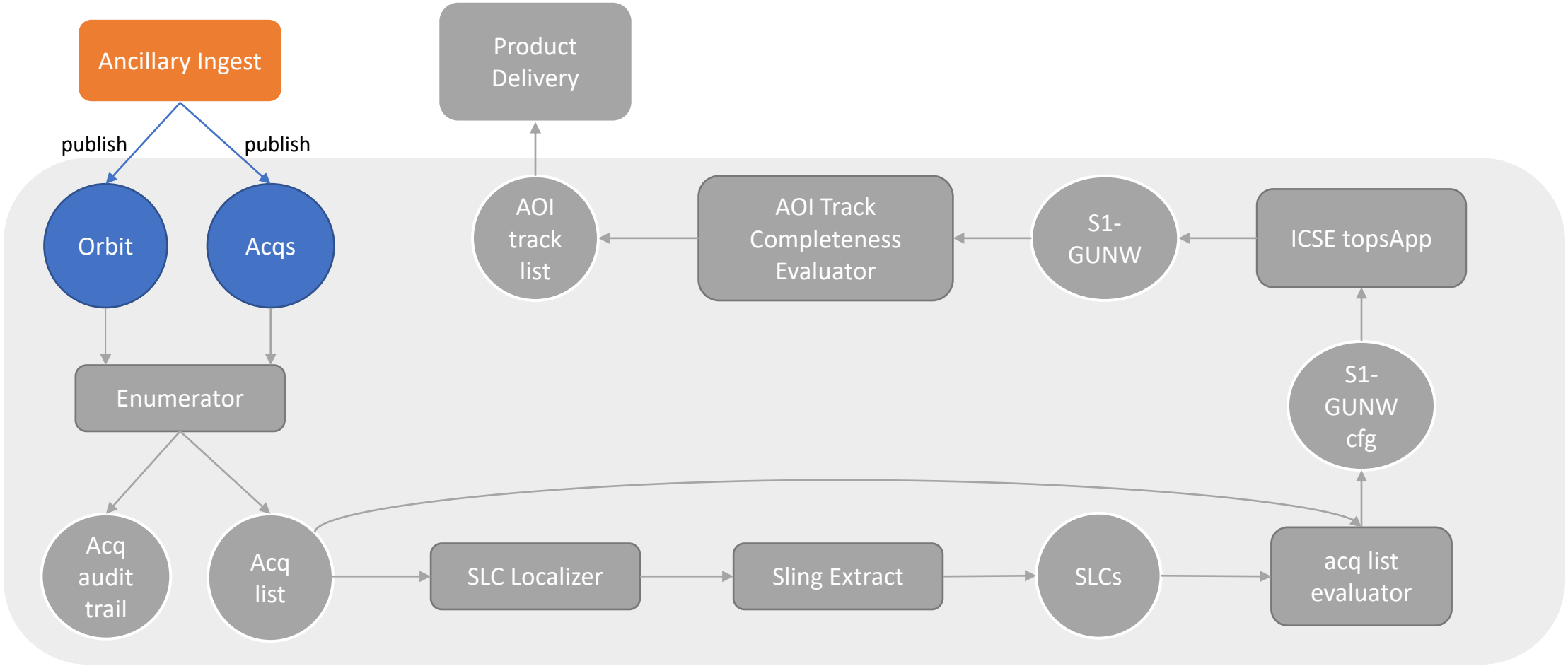
- Explicit workflow (SciFlo)
  - Control-flow patterns
    - execution order of individual steps in a workflow are explicitly defined
    - Sequence
    - Parallel split
    - Synchronization

- Implicit workflow (trigger rules)
  - Workflow data patterns
    - Event-based task trigger - external event initiates execution of an individual step in a workflow
    - Data-based task trigger – the evaluation of process data requirements evaluates to TRUE thus initiating execution of an individual step in a workflow

ARIA Standard Products Pipeline

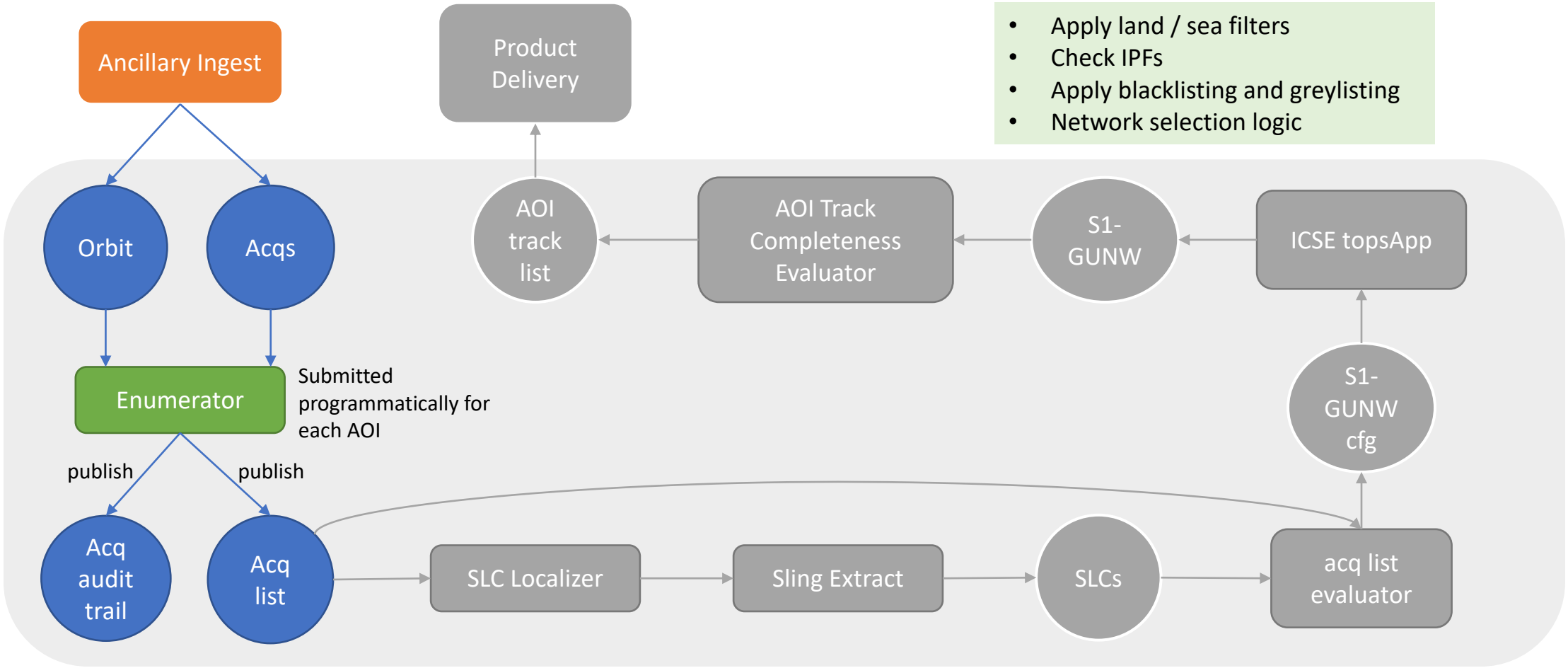
Jet Propulsion Laboratory
California Institute of Technology

S1-Geo Unwrapped Interferogram Production



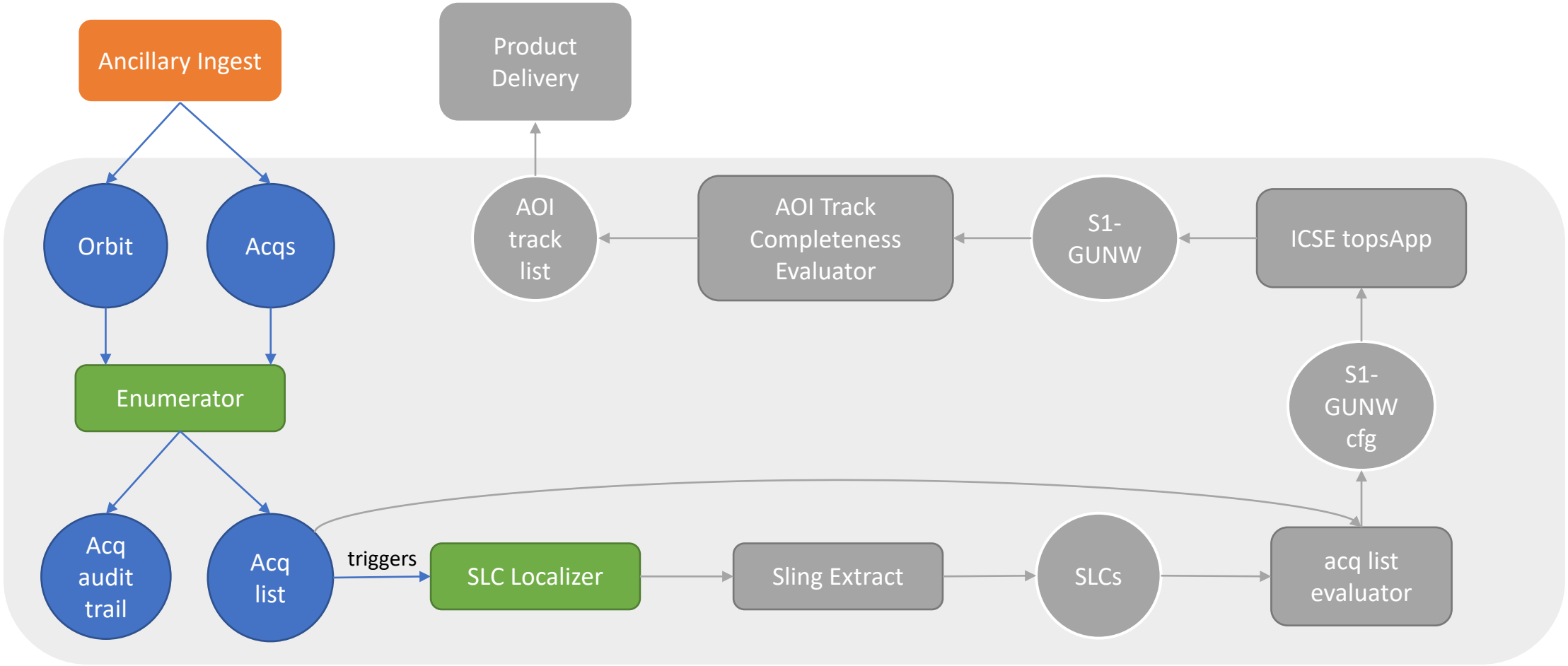
Jet Propulsion Laboratory
California Institute of Technology

S1-Geo Unwrapped Interferogram Production



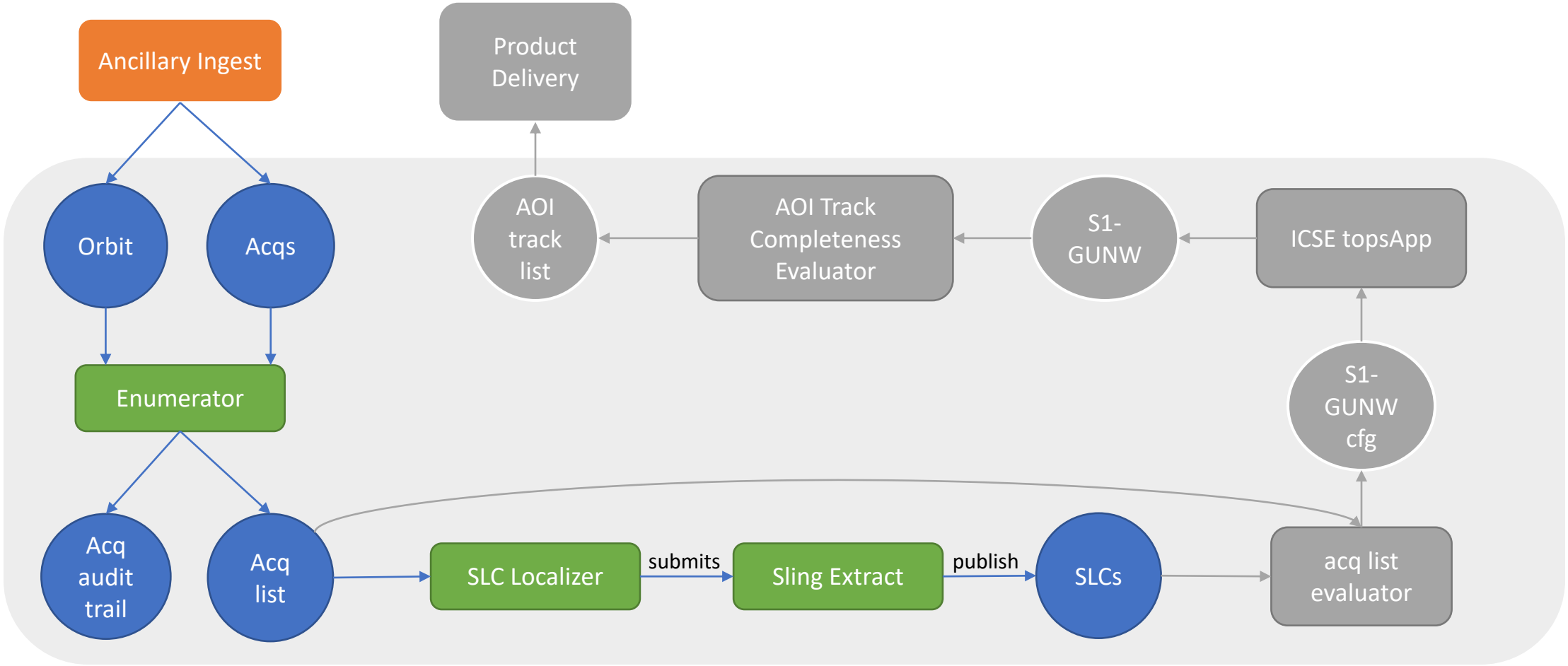
Jet Propulsion Laboratory
California Institute of Technology

S1-Geo Unwrapped Interferogram Production



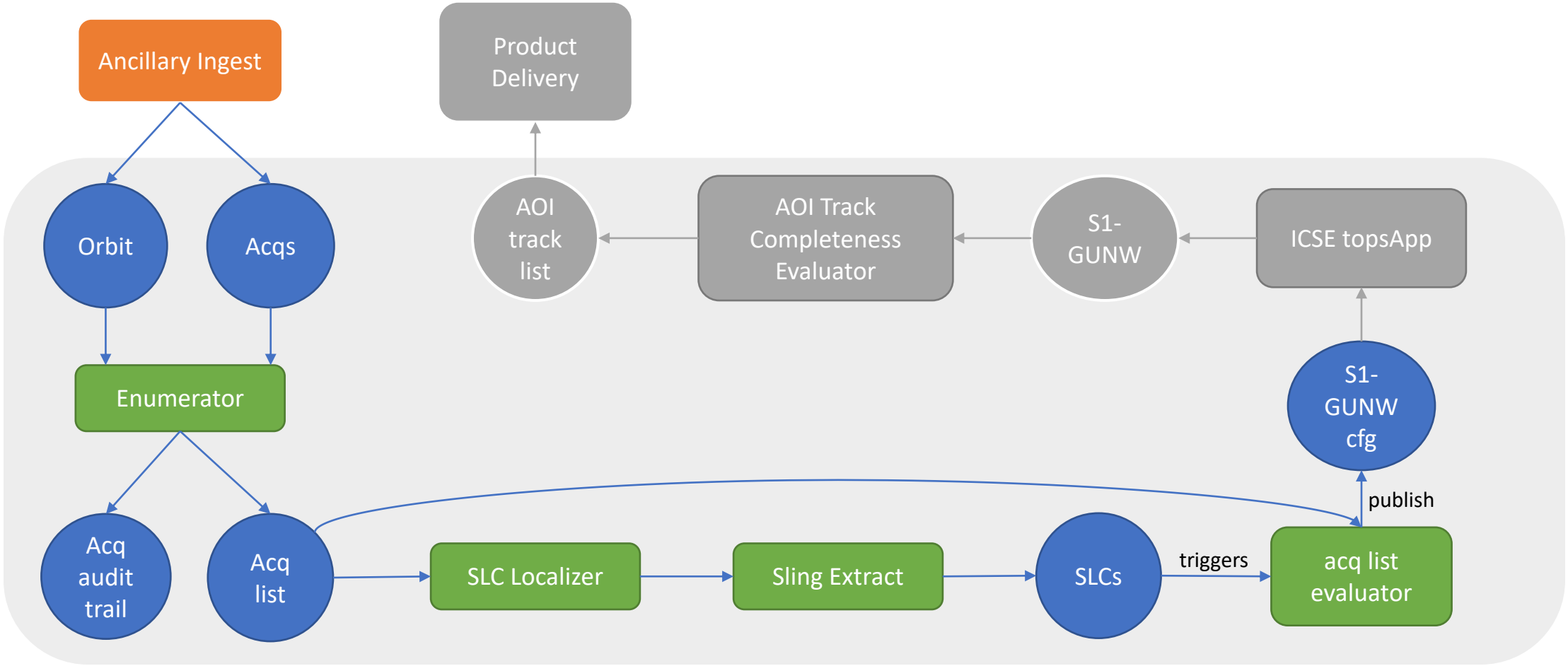
Jet Propulsion Laboratory
California Institute of Technology

S1-Geo Unwrapped Interferogram Production



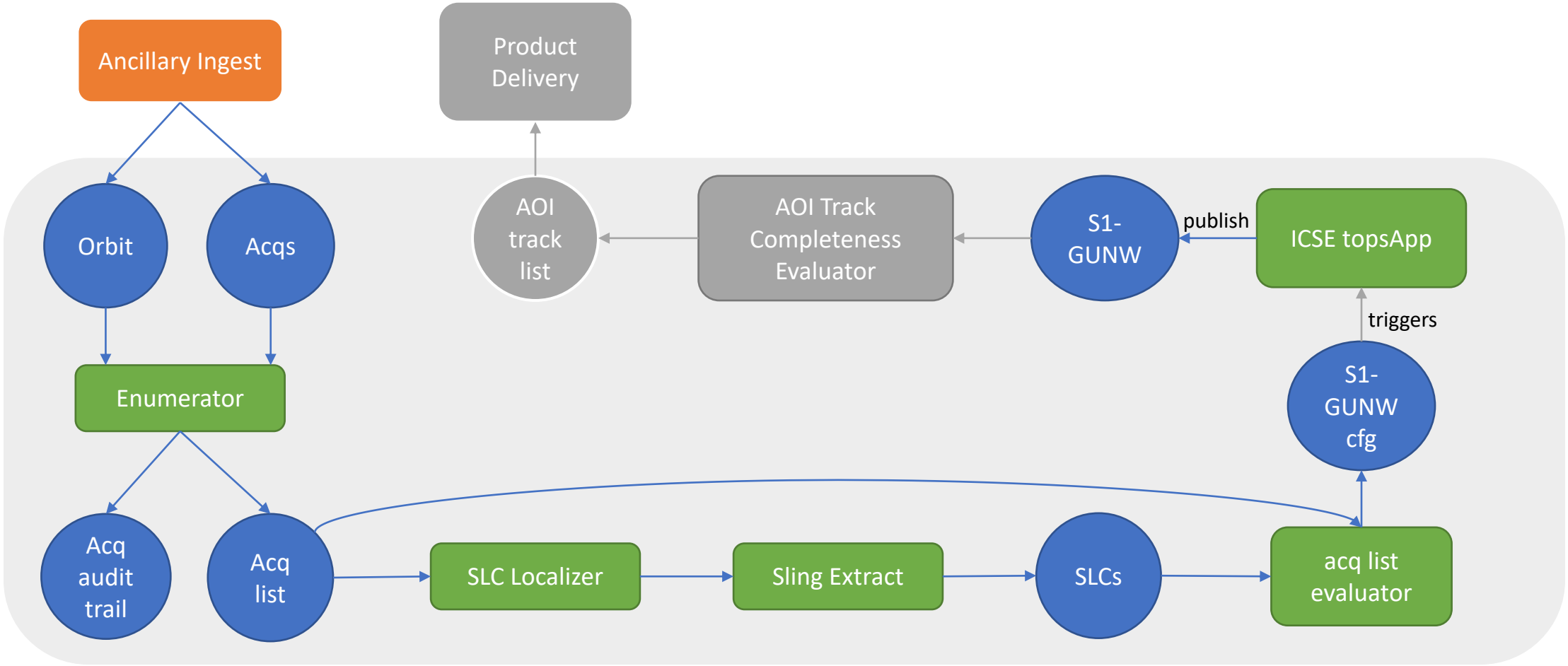
Jet Propulsion Laboratory
California Institute of Technology

S1-Geo Unwrapped Interferogram Production



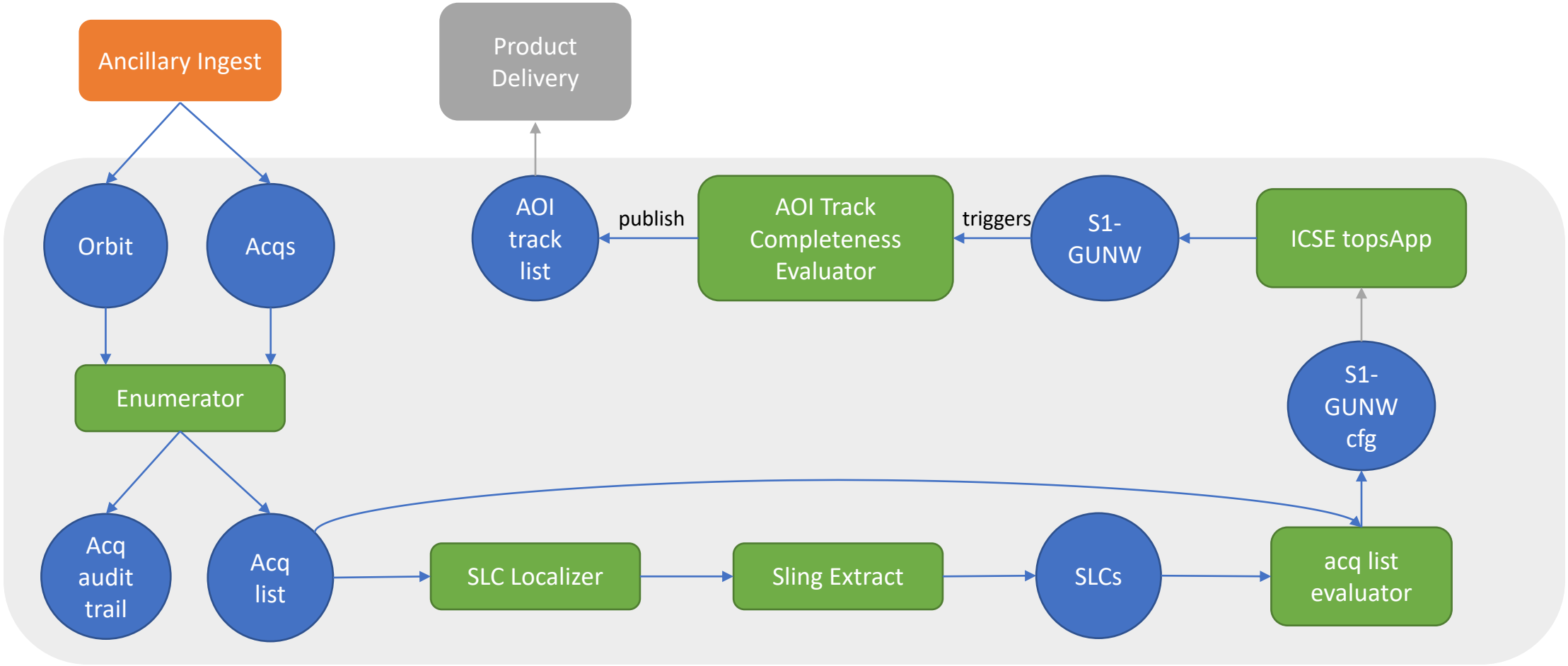
Jet Propulsion Laboratory
California Institute of Technology

S1-Geo Unwrapped Interferogram Production



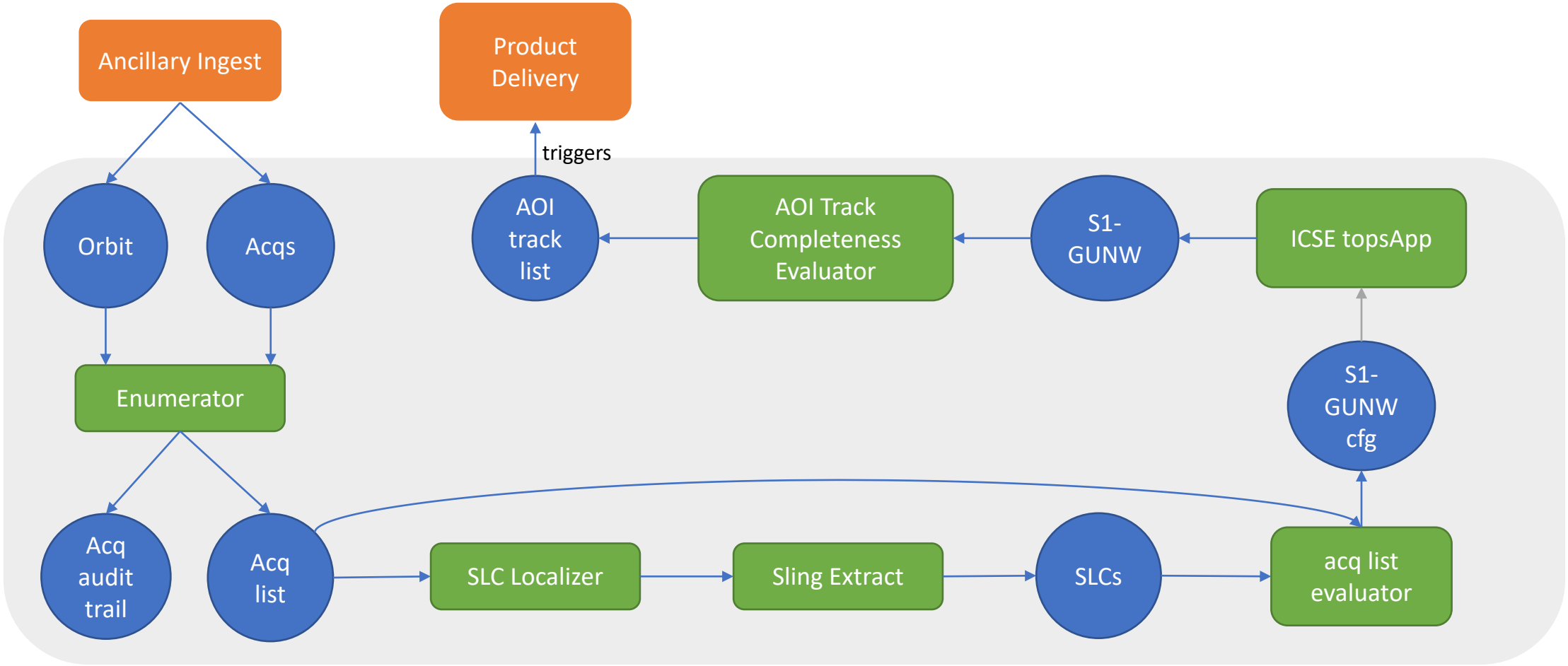
Jet Propulsion Laboratory
California Institute of Technology

S1-Geo Unwrapped Interferogram Production

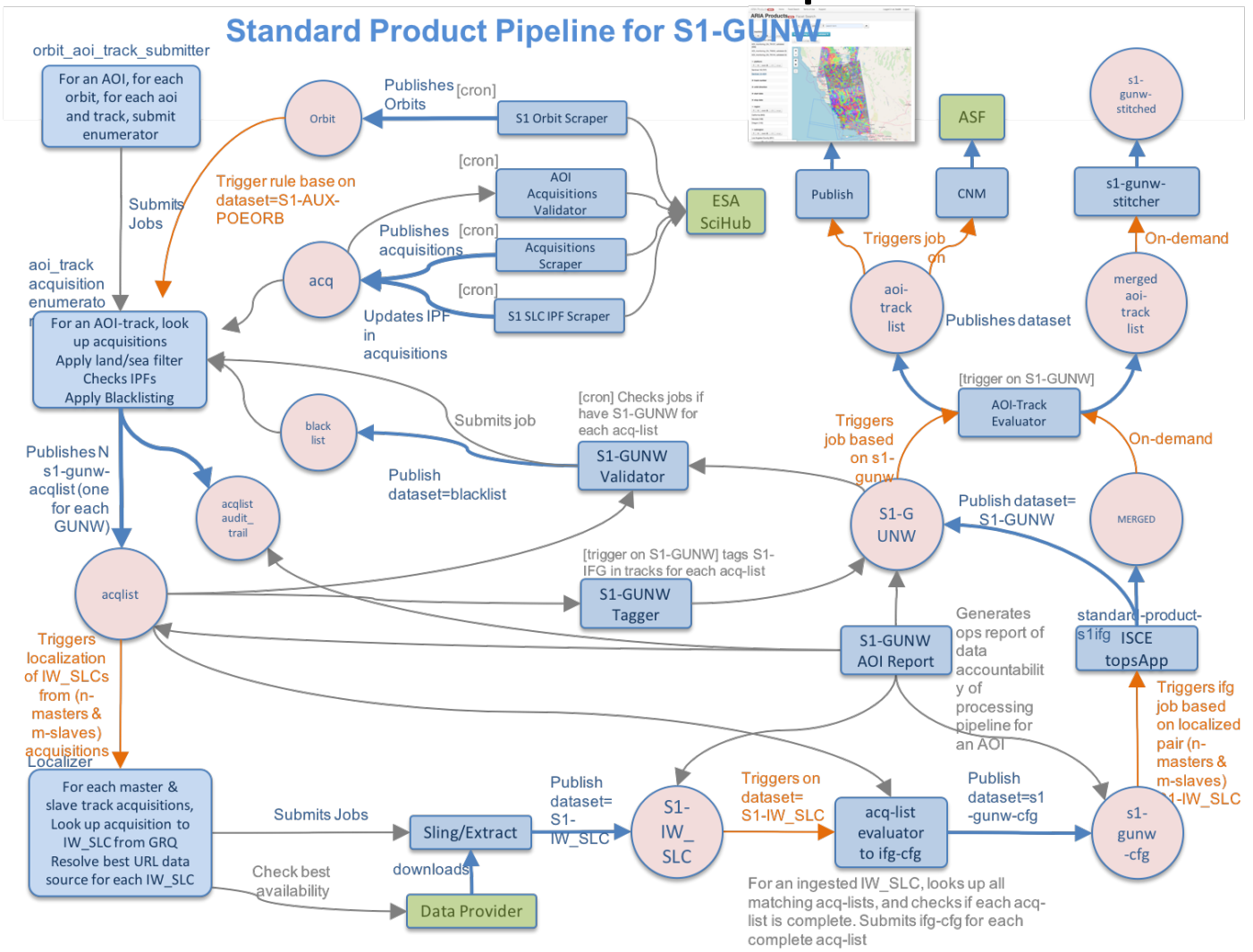


Jet Propulsion Laboratory
California Institute of Technology

S1-Geo Unwrapped Interferogram Production



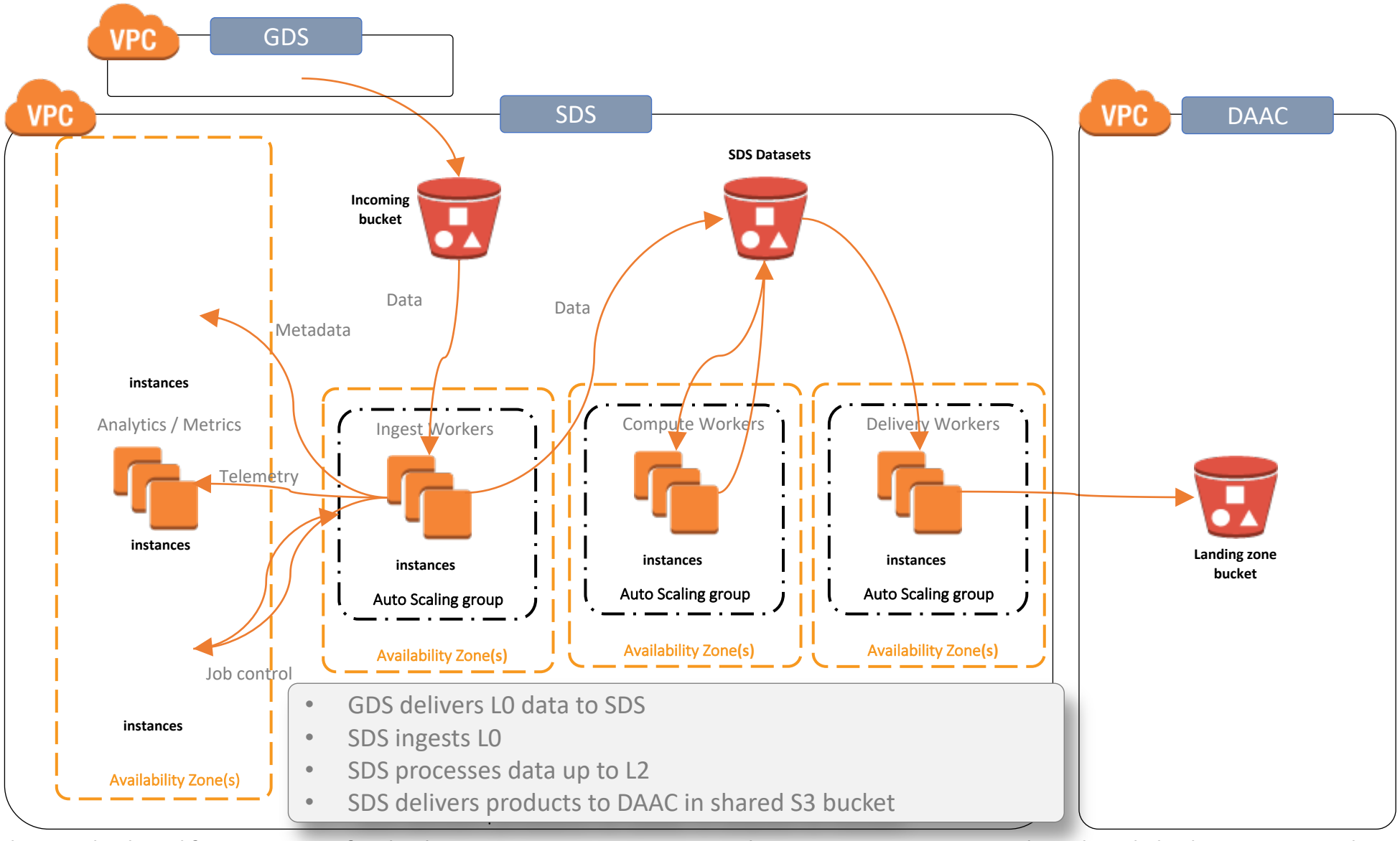
ARIA: Standard Products Pipeline



Mission Adaptation

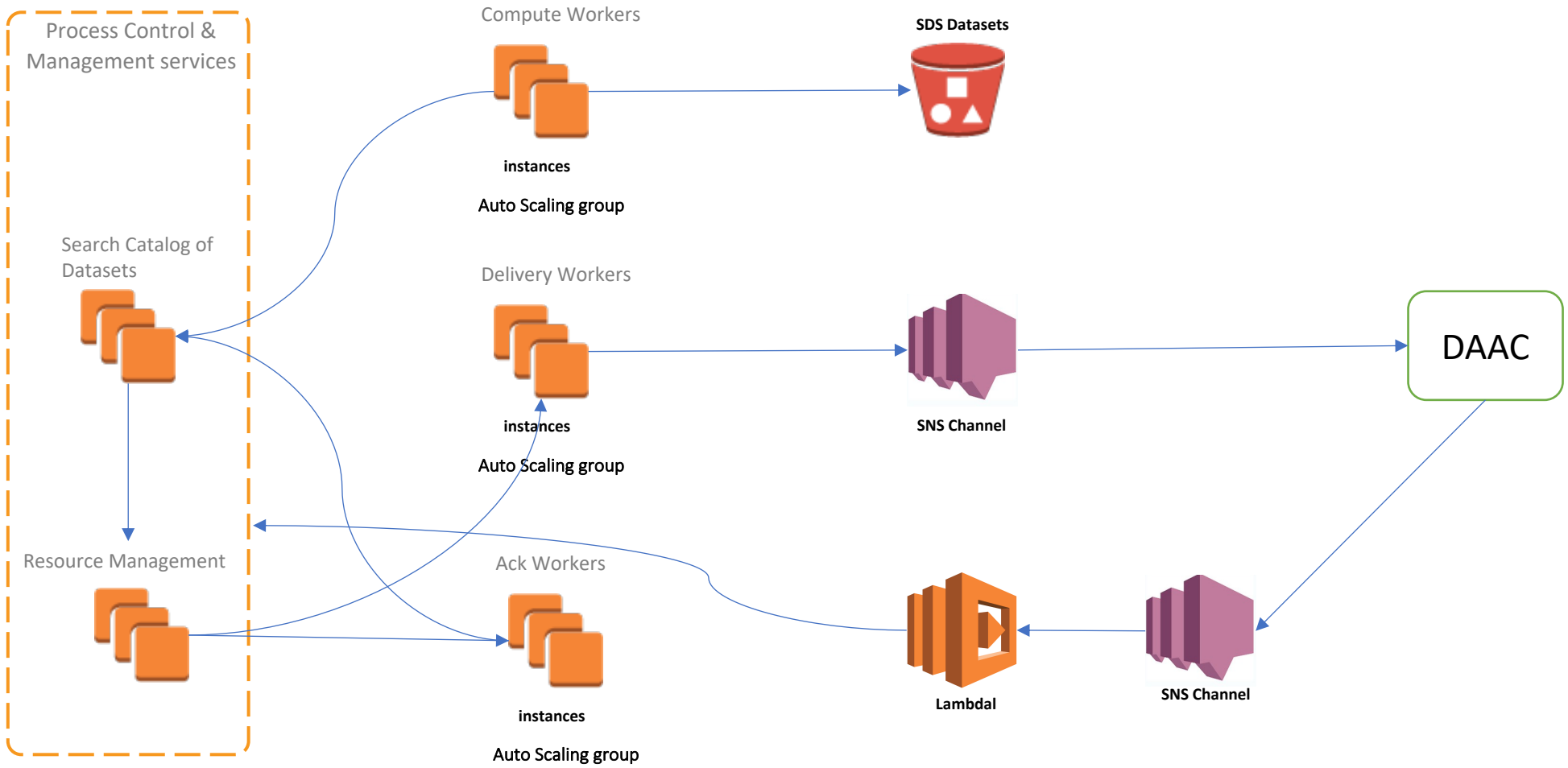
AWS Architecture

Jet Propulsion Laboratory
California Institute of Technology



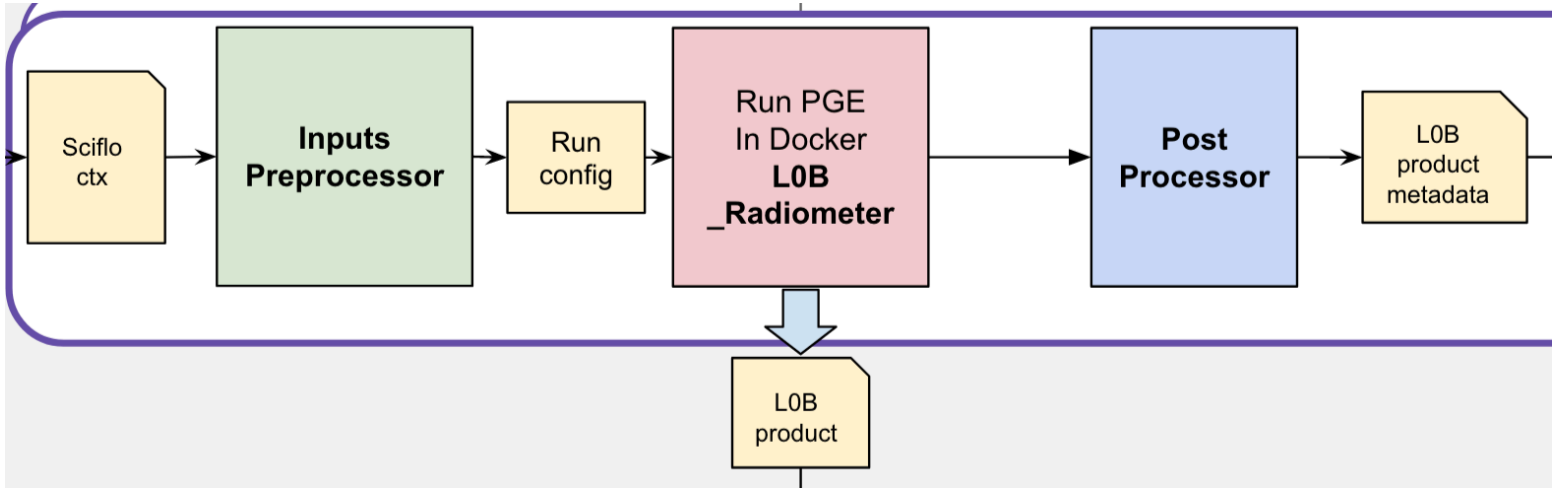
Jet Propulsion Laboratory
California Institute of Technology

Product Delivery Mechanism



SciFlo: Workflow Executor

- Generic framework for PGE Execution
  - Pre-processing: evaluate preconditions, query for proper input/ancillary files
  - PGE Execution on a remote worker
  - Post processing: pass metadata forward in workflow, trigger next steps or workflows



Challenges and Lessons Learned

Cost Optimization in Storage

- For ARIA
- Just-in-time downloading of IW_SLC
  - Cost savings in storage
  - Downloads from fastest data provider at that time
- Network enumerator now based on acquisition metadata
  - Cost savings in processing fewer scenes
- Rolling archive cache
  - Only store what is “hot”

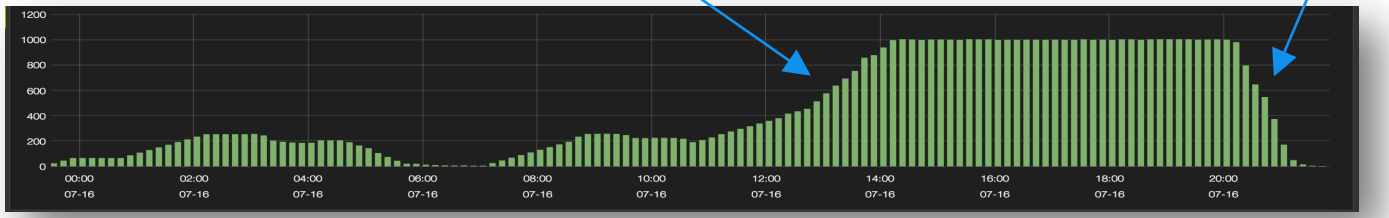
Optimizing for Scaling In/Out Events

Scaling up (scale out)

- Auto scaling group batching and timeout periods
- Scale up in group sizes of **multiples of availability zones (AZ) to m**inimize AZ load rebalancing terminations

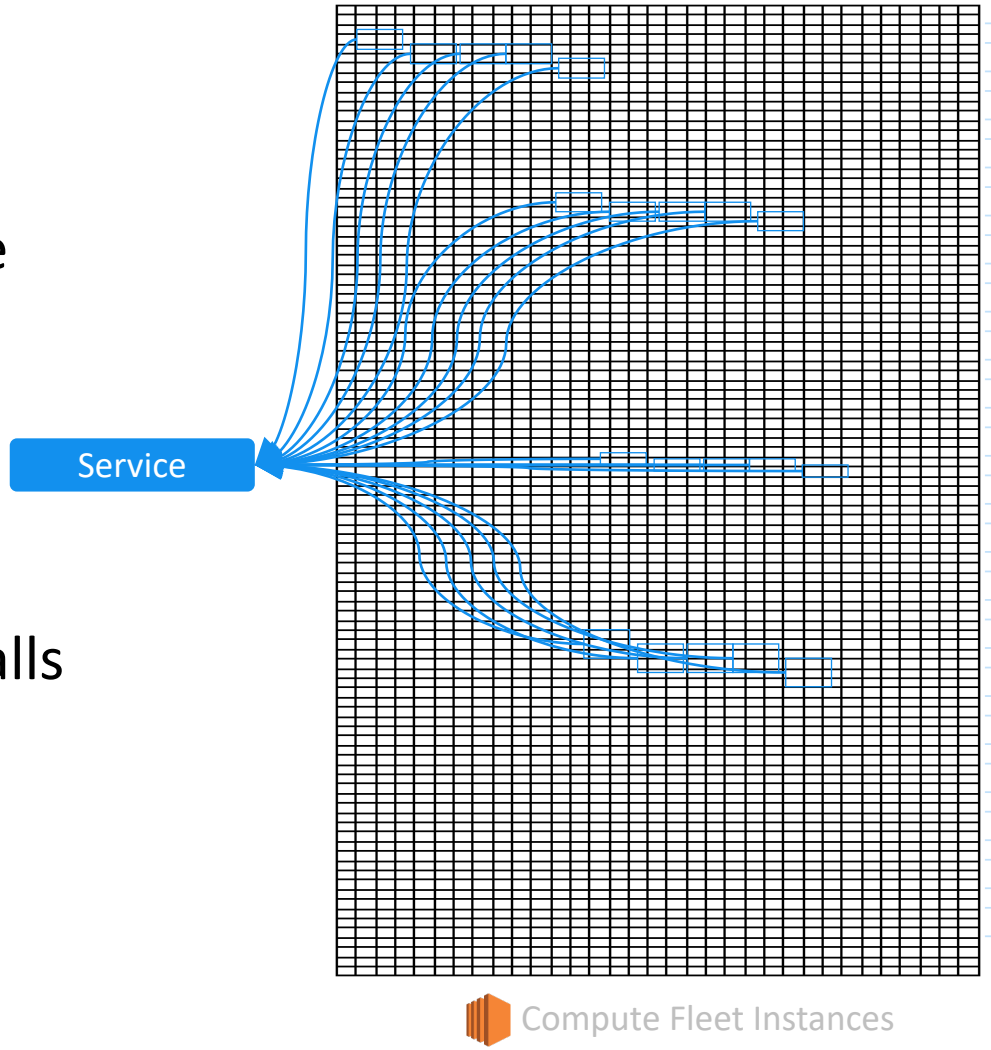
Scaling down (scale in)

- What policy to set to scale down?
  - E.g. CPU / network utilization
- Domain knowledge only known within the compute instances
  - Self-terminating instances
  - “harakiri” / “suppuku”



“Thundering Herd”

- Large fleet of auto-scaled compute instances calling same services at same time
  - *“API rate limit exceeded”*
- **“Jittering”** the API calls
  - Introduce ***randomizations*** to API calls
  - Distributes load on infrastructure



Open Source Project

- https://github.com/hysds
- https://github.com/aria-jpl

Extra Slides

Jet Propulsion Laboratory
California Institute of Technology

SWOT Processing Pipeline